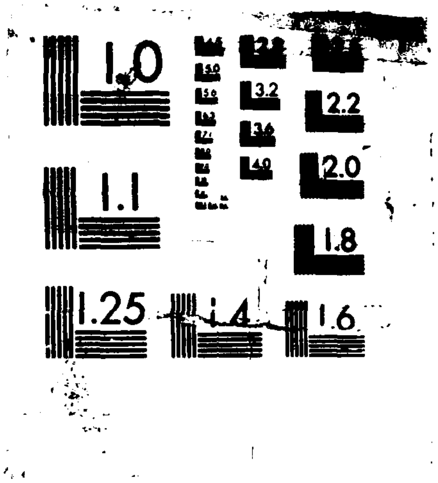
AD-A182 300 A STATIC RAM AS A FAULT MODEL EVALUATOR(U) STANFORD UNIV CA CENTER FOR INTEGRATED SYSTEMS J M ACKEN ET AL. 1987 MDA903-83-C-0335 1/1

UNCLASSIFIED F/G 9/1 NL



MICROCOPY RESOLUTION TEST CHART

A STATIC RAM AS A FAULT MODEL EVALUATOR

John M. Acken and Mark Horowitz

Center for Integrated Systems
Stanford University
Stanford, CA 94305

1987

Abstract

This investigation considers the relationship between the physical failures that occur during fabrication and the resulting faulty behavior of the circuit. Fault models are used to describe the operation of integrated circuits containing physical failures introduced during fabrication. The effectiveness of fault models is dependent upon both the accuracy of the model and the occurrence of the underlying failure mechanism. A specially designed static RAM provides the size, design style, and testability required of a fault model test bed. An actual RAM implementation is described which was used for evaluation of fault models.

Introduction

The increased size and complexity of today's integrated circuits (ICs) cause a very significant incr^ease in testing effort. The major research areas in custom IC testing include failure analysis, functional testing, test generation, fault simulation, and design for testability. Failure analysis is the determination of the cause of a failure. Functional testing, which includes all testing based upon logic values rather than parametric values, is used to detect faulty chips and to diagnose location and cause of the fault. Test generation provides the input vectors for functional testing. Fault simulation describes circuit behavior in the presence of a fault, and evaluates how well a given test set detects faulty circuits. Design for testability adds testing considerations to the design parameters of ICs. Fault modeling is common to all of these areas. Until now, most studies in each of these areas have have been isolated from each other. Fault models, used to evaluate functional tests and design for testability are theoretically derived without emphasis on physical failure mechanisms. A vehicle is needed to provide the connecting fibers between the areas of testing by presenting a direct correlation between the causes of failures and the fault models used for testing and design.

The mechanical or chemical action that actually causes undesirable operation of an integrated circuit is termed a **physical failure mechanism**. The resulting damage causes the failure mode. Failure will be used in this paper to encompass both these ideas and to represent the actual problem with the circuit. The incorrect operation of an integrated circuit containing a failure is represented by a fault model. Fault models may describe logical or electrical misbehavior of an IC. Figure 1 shows examples for each of these definitions. The accuracy of the fault models limits all analysis based upon those models. Therefore, some method of evaluating the accuracy of fault models is needed. A carefully designed static RAM provides an excellent test bed for performing this evaluation.

Circuit operation in the presence of a failure is very technology dependent. Therefore, fault models which attempt to accurately reflect reality must be matched to the technology. The technology discussed in this report is CMOS with two layers of metal, but the general aspects of utilizing a RAM for evaluation of fault models is applicable to any technology.

The goal of a fault model is to describe the behavior of a circuit in the presence of a failure. In order to sensibly model the faulty behavior of a circuit the underlying physical cause or failure must be considered. Physical failure mechanisms can be categorized based upon the length of time between manufacture and when the failure affects the operation of the circuit. Those failures which immediately prevent a circuit from operating are referred to as yield limiting failures. Those failures which affect the circuit after a period of operation are termed reliability failures. The RAM as a test vehicle approach can be applied to either of these categories. However, this report will concentrate on evaluating fault models for yield limiting failures.

Fault models provide the underlying groundwork for many testing concepts. Since integrated circuits contain physical failures, the fault models must be evaluated with respect to the failures. A static RAM conforming to custom logic design practices provides a method for measuring the effects of failure mechanisms.

Characteristics of a Fault Model Test Vehicle

Testing and fault modeling must be based upon the physical failures that cause circuits to malfunction. The question has frequently been asked, "What

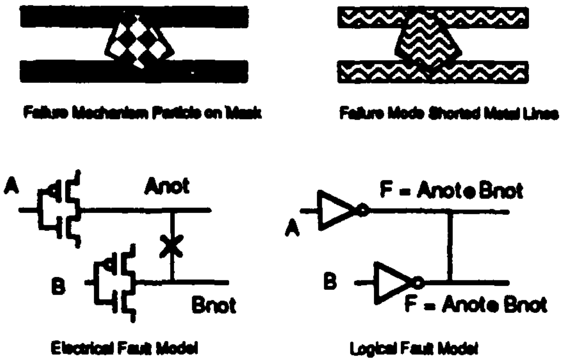


Figure 1: A Failure Mechanism, Failure Mode, and Fault Model Example.

physical failures would really cause the behavior described by a given fault model?" However, to be more complete the question should be, "What fault models can simulate the behavior of chips containing physical failure mechanisms?" To answer this question, a method for deriving fault models describing the behavior of chips containing failures is needed.

The analysis of actual failure mechanisms as they relate to practical fault models places special requirements on any test vehicle. First, the test vehicle must be large and complex enough to both represent actual IC designs and provide a statistical likelihood of failures. Since the motivating factor for improved modeling is the increase in size and complexity, the evaluation of fault models on small test vehicles avoids the very problem driving the need for better fault models. Also, since the goal is to derive fault models based upon failures, some failures must be present. Since yield is a function of size, the test vehicle must be large enough to have a reasonable number of failures for each fault model. Second, the test vehicle must be typical of both the IC design and the IC fabrication process to avoid introducing exotic failures or eliminating typical failures. The various steps within the complex process of IC fabrication sometimes have a strong interaction with each other. Any test vehicle must evaluate all of the interactions in the presence of a fault. Third, the test vehicle must be very testable to allow both detection and isolation of failures and the resulting electrical and logical effects. A test vehicle which masks and hides failures is not valuable for examining the failure's effects upon circuits. Just as ineffective is a test vehicle which indicates the presence of a failure but does not allow isolation and analysis of the failure. The effort to test the vehicle should be minimised to allow more effort to be applied to evaluation of various fault models for physical failures. Test vehicles for fault models should be evaluated based on the three criteria of size, design generality, and testability.

Several methods exist for generating and evaluating fault models. The first method is theoretical definition, yielding, for example, the classical stuck-at model. The goal of such models is to provide a framework for evaluating the behavior of designs and tests in the presence of faults. This method does not meet the criterion of size, since the analysis does not consider all of the many possible cases. The most serious drawback of this method is having to ensure that significant failures are not forgotten and that interesting but nonexistent failures are not included. The strongest argument for theoretical analysis is testability, that is, all conditions are stated, therefore totally controllable, and all values can be derived. But all of this analysis leads to a rationalization of a given model looking for causes, rather than finding the causes and trying to model the effects.

The most common method for evaluating fault models is simulation. For example, a logical fault model for a bridging fault (a short between two nodes) can be analyzed at the circuit level with SPICE [1][2]. In order to satisfy the first criterion of a good test vehicle for fault models, the simulation approach requires a vast amount of computer time and human effort to ensure an accurate comparison of the likely failures. The effort includes many simulations of many circuit variations and extensive modeling for both good and faulty circuits. Satisfying the second criterion includes exhaustive searches for types of failures and evaluations of the relative occurrence of types of failures. Simulation fulfills the third criterion since all of the information about the circuit is available, within the limits of modeling accuracy. Another approach is to "construct" a failed circuit and measure the results [3]. While this approach guarantees the existence of failures that the first criterion relies upon size to provide, it does not provide the variety of circuits that a complex test vehicle does. The requirement to include all but only typical failures involves the same searches and evaluations that simulation does, but it also raises the question of the effects and general applicability of induced failures. This method can be especially good in meeting the testability criteria, since the failure is known and can be readily compared to the operation of a good circuit.

Another approach is to design a specific test structure. The trade-offs between size and testability be-come very apparent while trying to design a test structure. Large structures are likely to have failures present but the very size limits access to the particular failure site. For example, measuring transistor characteristics requires access to all terminals of the individual transistor, but that typically requires many pads. A transistor array of 200 transistors was designed by the author in such a way that only 20 probe pads were required. However an array of 200 transistors is orders of magnitude short of the number needed to ensure the presence of some failures. Also the design had various sizes of transistors, but this fails the criteria for excluding exotic failures should they be size dependent. And for all designed test structures the question of standard design procedures haunts the project. None of these methods fully satisfies the need for a test bed for deriving fault models based upon physical failures.

One approach which does meet the above three criteria is to use a large, complex, typical, but highly testable circuit. A specially designed static RAM meets all of these criteria very well. The first criterion requiring a large sample is readily met by adjusting the size of the RAM. RAMs are real circuits so they avoid the need to evaluate the applicability of the failures being considered. However, RAM design and failure analysis is typically very dependent upon analog circuit behavior. To maintain suitability for fault model derivation, a RAM must be designed to avoid dependence upon special analog behavior. This is achieved by using a static ram, with simple sense amplifiers, and designing with exactly the same design rules and constraints that are used for custom logic. The third criterion of testability is met since RAMs perform a very simple logic function, with all states controllable and observable.

A Particular RAM as a Fault Model Evaluator.

This section describes a particular design for a RAM used as a test bed for fault models. This RAM has a dual purpose for its existence, one as a test vehicle for analyzing fault models and failure mechanisms, and another as an internal instruction cache for the MIPS-X microprocessor[4]. The second purpose actually enhances the circuit's value as a test bed for fault models because it utilizes the same design constraints as the microprocessor portion of the chip. Some of the unique design decisions in this RAM were driven by the second application — but were applied to the design in such a way as to maintain suitability as a test vehicle.

Multiple fabrication runs were used to make RAM test vehicles. One technology for this RAM is the MOSIS 3μm, two level metal, p-well CMOS. There is also a 2 μm two level metal n-well CMOS version manufactured at Stanford and MOSIS.

A typical RAM is asynchronous, however, since this design is intended for a specific application, it is a synchronous design with 2 non-overlapping phases of the clock. The organization of the RAM is 256 words of 32 bits per word (the 3 μm version is 256 words by 24 bits). Many pads were added to ease analysis of failures. Several test probe taps have been added to detect, diagnose, and further analyze any faults present in the circuit. Seven pads are devoted to the Vdd and ground distribution to measure voltage shifts throughout the RAM. Two word lines are observed through output pads allowing external measurements of word line activity. A minimum size inverter is connected to the wordline to allow measurements without significant loading on the word lines.

This RAM uses the two phase non-overlapping clock scheme common in custom logic design. The full clock cycle is 50 nanoseconds. The RAM precharges during $\Phi 1$. The write signal must be stable $\Phi 2$, and the address is valid $\Phi 1$.

A block diagram for the RAM is shown in Figure 2. Primary modeling is based upon faults in the memory array, however the rest of the RAM also provides potential failure sites. The address latches are flow-through latches, allowing the address to percolate through the nand gate predecoders into the address decoders. A word line, when selected by the decoders, turns on the two access n-channel pass transistors for each RAM cell. A given address raises two word lines, one on each side of the RAM, with 128 cells accessed by each word line. The bit select circuitry is driven by the 8 tag lines. For a read cycle, the bit lines drive the sense amplifier, which in turn pulls down the data bus that was precharged during $\Phi 1$. For a write cycle, the bus drives the pair of inverters that make up the bit line write drivers. The write drivers discharge the bit or bitbar lines through the bit line select, which writes the RAM cell.

The RAM Cell

This design uses a six transistor fully static CMOS RAM cell. This circuit more accurately reflects circuitry in custom logic designs than any other type of memory cell. The RAM cell utilizes full CMOS inverters for feedback to store data rather than special capacitors as in DRAMS or polysilicon pull-up resistors. Also, just as the most common gate in a custom logic IC is an inverter, the most common gate in this RAM is an inverter. The access transistors are

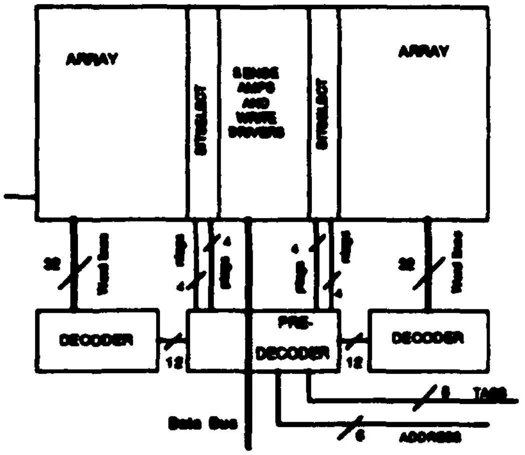


Figure 2: The RAM Block Diagram

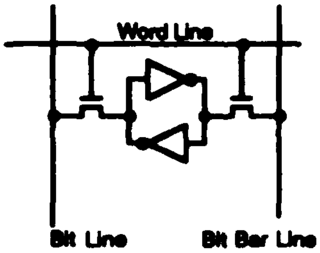


Figure 3: The RAM Cell Schematic

n-channel pass transistors which are adequate since all reading and writing for the cell is accomplished by pulling one bit line low. Bit lines only need to be pulled low because both bit lines are precharged before the read/write cycle. Precharging bitlines helps analyse the effects of failures on precharged logic. The circuit schematic is shown in Figure 3.

Sense Amplifiers

The schematic for the sense amplifiers is shown in Figure 4. The sense amplifier inputs are precharged and equalised on Φ1, and the internal nodes are predischarged. All transistors in the sense amplifier are turned off as the bit lines begin to change. As one bit line or the other begins to drop one threshold (V_{th}), the p-channel transistor at the top of the sense amplifier begins to source current to the midpoint of the other side of the sense amplifier. This turns on the pull down transistor which keeps the midpoint of the leg opposite the dropping bit line low. If a 0 is being read, that is, bit rather than bitbar goes low, the large n-channel transistor discharges the data bus. If a 1 is read, the bus remains high and the other large n-channel transistor provides a balancing load. This simple level sense amplifier maintains compatibility with custom logic without introducing unacceptable delay. The path from the bit line through bit decode through sense amplifier to bus has a delay of 7 - 8 ns.

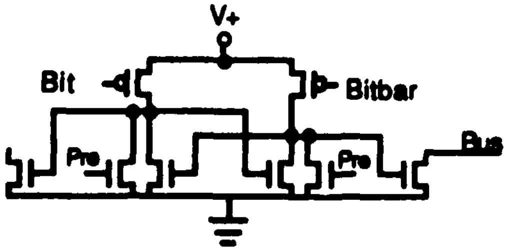


Figure 4: The Sense Amplifiers

Other Circuits

In addition to the inherent testability of the RAM, some circuitry was added to observe internal nodes. To aid in diagnosis of decode logic or timing problems, each of the wordlines was provided with a 10 μm by 10 μm Metal 2 pad for probing. Also, several internal signals were connected to output pads to allow timing analysis. Since the RAM is specifically designed to be typical of custom logic IC design, the RAM used exactly the same design rules and circuits used in custom chips rather than special rules and circuits use for memory design.

Testing Results for the RAM

The testing of the RAM demonstrated the correctness of the design, the usefulness for detecting failures, and a similarity of RAM results with industry results. Five batches of 3 μm RAMs have been tested. All of the batches were from the same design (identical masks) but from different fabrication lines. The first batch had about one half completely working parts and the other half had a variety of failures. All of the chips in the second and fourth batches failed even simple tests. The third batch only had 2 chips bad out of 13. The fifth batch had 3 completely working chips out of 18 tested. Figure 5 shows the distribution of failing chips.

Excluding visual inspection and test structure measurements, the detection of faulty ICs occurs in three steps. The chips that draw excessive power supply current or that fail on the first few test vectors are marked at the first step. The second step is to apply a moderate number of test vectors and exercise the major functions of the chip. The third step is extensive functional and rigorous parametric testing. The extensive functional test involves a large test vector set intended to exercise all of the internal circuitry. The

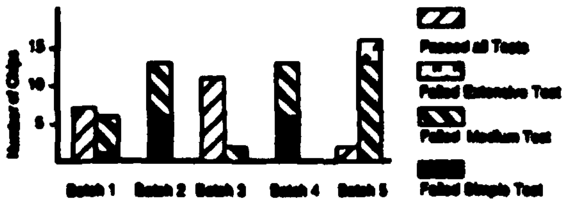


Figure 5: The RAM testing Results.

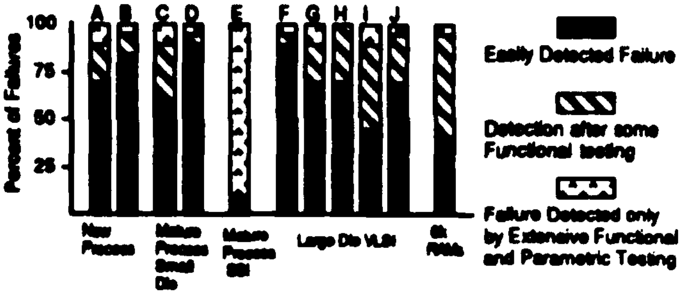


Figure 6: Distribution of Failures Detected During Testing.

rigorous parametric tests include speed testing, current drive, and voltage levels on external pins. The testing of the RAM was performed in the same three basic steps. The first step was a very simple test including power supply current, address line taps, and some simple internal control line values. The second step was extensive testing for simple functions. The third step was exhaustive RAM tests and diagnosis of existing failures. When chips failed they usually failed catastrophically in the first step, or at the latest during the second step of testing.

About half of the chips passed all levels of testing, but almost all of the failures had been detected before the final exhaustive testing. In fact, about half of the failures occurred in the first simple test step. This corresponds with industry experience. Figure 6 shows how several companies, and some divisions within two companies, describe the distribution of failures during testing. These numbers do not include any yield numbers. However, the data for bars e and j were both noted as particularly high yield product lines. The companies include some large semiconductor companies as well as research laboratories. Universities are not included, since testing data is not centrally collected. The general trend shows large VLSI and new processes tend to have more general failures that show up earlier in the testing procedure, but that mature fabrication tends to have most of the failures discovered only after very exhaustive testing. This matched our measured result closely since this is a VLSI chip submitted for manufacture using the relatively new MOSIS 2 μm technology. Also, since batches of chips from MOSIS do not come from the same manufacturer, it is worth noting that this rule applies to all — most failures are readily detected on complex chips manufactured on new processes, whereas the random, singly occurring faults dominate the small chips manufactured from mature processes. The significant quantities of easily detected failures in both the RAM and IC fabrication as a whole supports the idea of using this approach to derive fault models caused by physical failures.

Summary

A method for deriving fault models based upon the underlying physical failure mechanisms is required for improved model accuracy. An experimental device for deriving fault models must be large enough to be realistic, standard enough in design to avoid introducing or excluding failure mechanisms, and testable enough to be useful in evaluating the fault models. A static RAM conforming to custom logic design practices is an excellent test vehicle for the derivation of fault models. A particular in.plementation of such a test vehicle has been fabricated (both at MOSIS and at Stanford) and tested.

Acknowledgements

The authors wish to thank Edward J. McCluskey, Steve D. Millman, Henk A. Goosen, and Gregory G. Freeman for their helpful discussions. The authors also wish to thank DARPA for support under contract MDA903-83-C-0335 and both the MOSIS project and ~~the Stanford IC Lab for~~ fabrication support.

References

[1] G. G. Freeman. *Development of Logic Level CMOS Bridging Fault Models*. Technical Report CRC 86-10, Stanford University, July 1986.

[2] C. Timoc, M. Buehler, T. Griswold, C. Pina, F. Stott, and L. Hess. Logical Models of Physical Failures. In *1983 IEEE Test Conference*, 1983.

[3] J. Soden and C. Hawkins. Test Considerations for Gate Oxide Shorts in CMOS ICs. *IEEE Design and Test*, 56-64, August 1986.

[4] M. Horowits, J. L. Hennessy, P. Chow, P. G. Gulak, J. M. Acken, A. Agarwal, C-Y. Chu, S. A. McFarling, S. A. Prsbylski, S. E. Richardson, A. Sals, R. T. Simoni, D. C. Stark, P. A. Steenkiste, S. W. D. Tjiang, and M. J. Wing. A 32b microprocessor with on-chip 2k byte instruction cache. In *Intl. Solid State Circuits Conf.*, February 1987.

END

8-87

DTIC